TABLE NO | | | |

STUDENT ID NO | | | | | | | | | | |

MULTIMEDIA UNIVERSITY

SUBJECT CODE _____

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2019/2020

TSD 2711 – Software Design

(All sections / Groups)

23 OCTOBER 2019
9:00 am – 11:00 am
(2 Hours)

Question	Mark
1	
2	
3	
4	
5	
Total	

Examiner 1 Signature: _____

Examiner 2 Signature: _____

Examiner 3 Signature:

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 13 printed pages (including cover page).
2. Attempt **ALL** questions. The distribution of the marks for each question is given.
3. Please write all your answers **CLEARLY** in the specific answer box provided for each question. Submit this question paper at the end of the examination.

Question 1 (10 Marks)

The software that supports a computerized banking network called MMU BANK. The network enables customers to complete simple bank account services via automated teller machines (ATMs) that may be located off premise and that need not be owned and operated by the customer's bank. The ATM identifies a customer by a cash card and password. It collects information about a simple account transaction (e.g., deposit, withdrawal, transfer, bill payment), communicates the transaction information to the customer's bank, and dispenses cash to the customer. The banks provide their own software for their own computers. The MMU BANK software requires appropriate record keeping and security provisions. The software must handle concurrent accesses to the same account correctly.

Figure 1 shows the Use Case diagram for the above scenario.

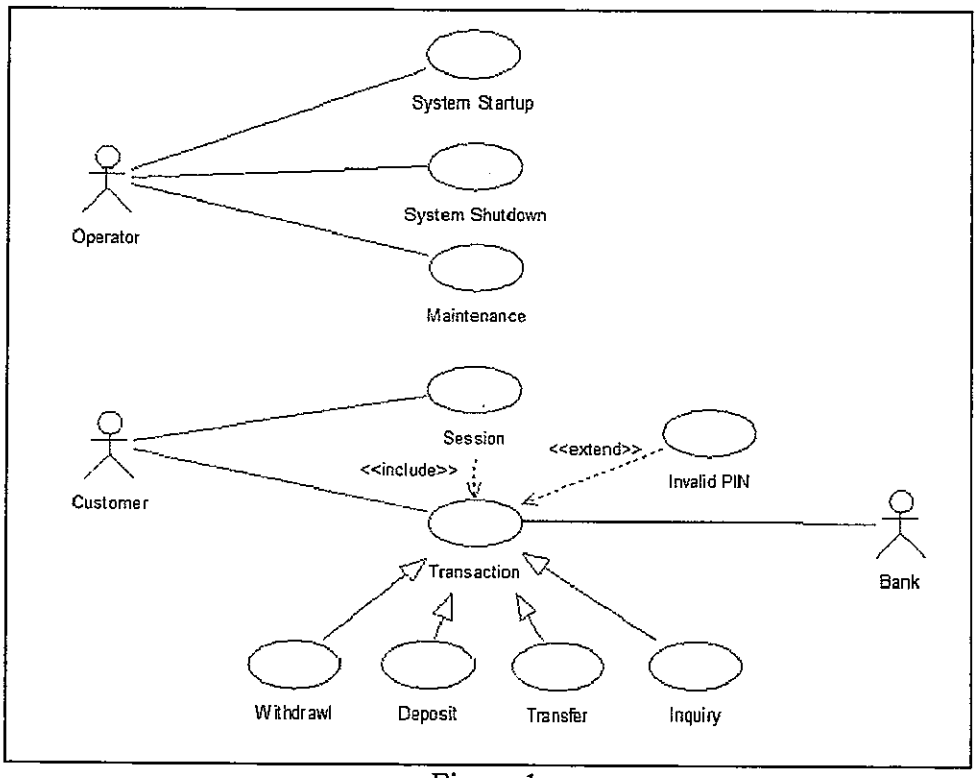
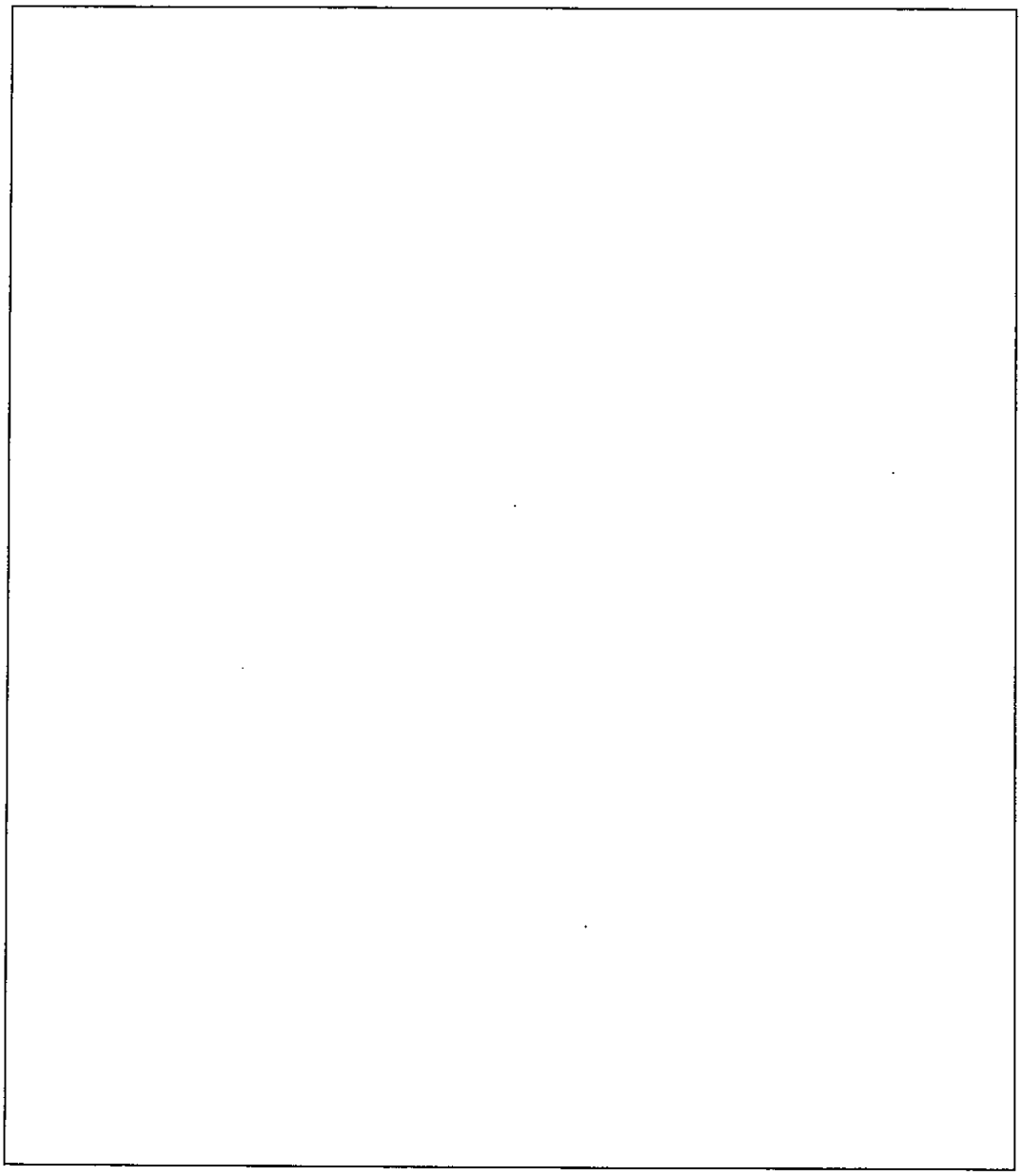


Figure 1

Continued...

a. Design a class diagram based on the scenario and use case above.

(8 Marks)



Continued...

b. What is the main purpose of use case modeling?

(2 Marks)

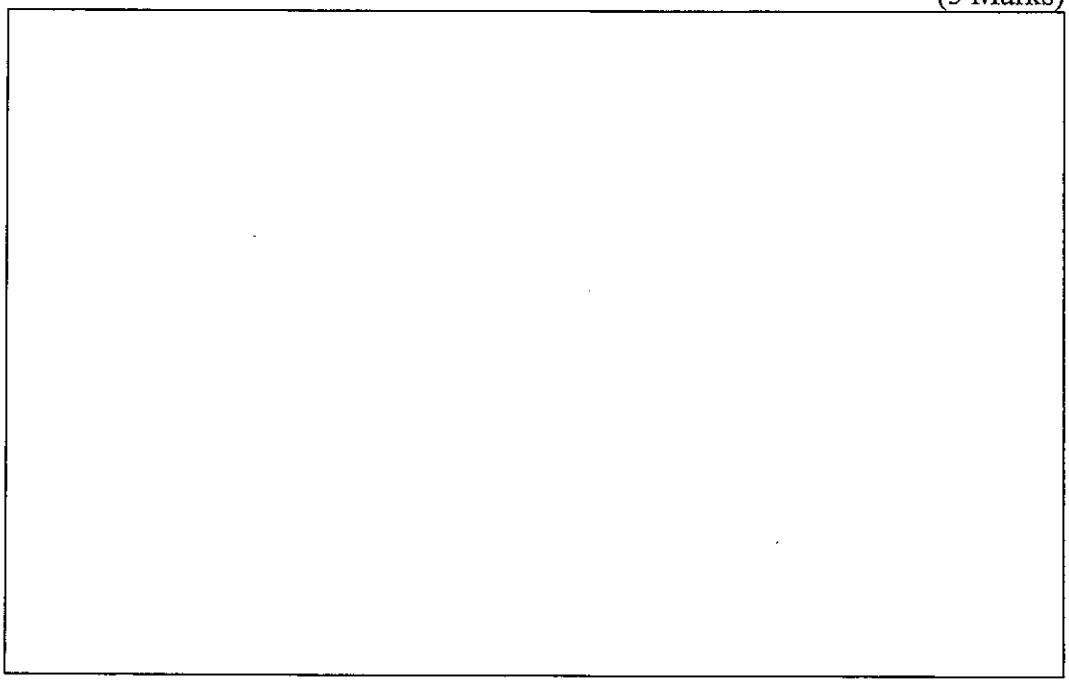
Continued...

Question 2 (10 Marks)

You need a program that calculates the checkout total for a customer at a cash-and-carry. To shop at this cash-and-carry, you need to be a member and memberships come in three types: bronze, silver and gold. Each membership type gets its own discount! A customer at a checkout first gives details of their membership card. The checkout assistant then scans in each item. The price is retrieved from the stock database. All the items are added up and the appropriate discount subtracted from the total. The checkout assistant then tells the customer how much is due. They receive cash from the customer, then calculate and give out the change as well as an itemized receipt.

a. Based on the scenario above, draw an Entity Structured Diagram

(5 Marks)



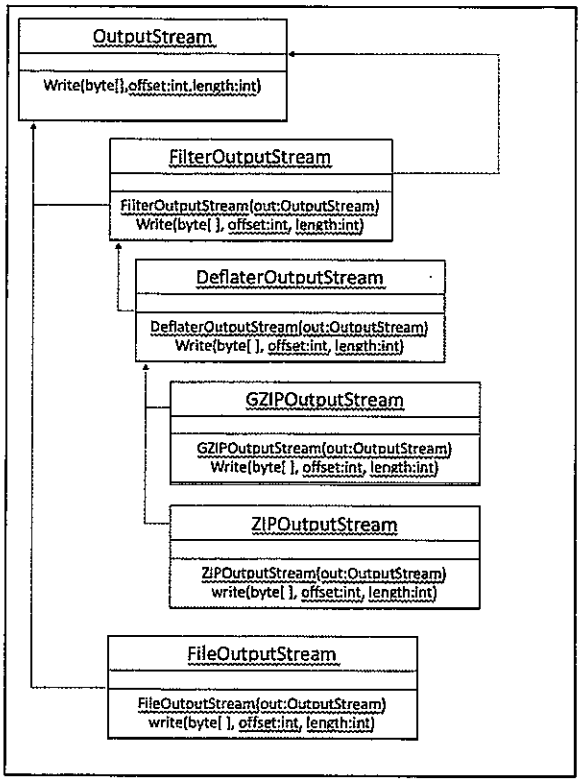
Continued...

b. State the goals for the software architecture and modularization for the web system. **(5 marks)**

Continued...

Question 3 (10 Marks)

Consider the following class diagram.



a. Identify the design pattern that is shown in the diagram , and explain how the design pattern functions (by referring explicitly to the elements shown in the class diagram). **(5 Marks)**

Continued...

b. Abstract classes and interfaces are the main enablers of software design patterns. Write down three differences between abstract classes and interfaces.

(3 Marks)

c. What does an interaction diagram depict? **(2 Marks)**

Continued...

Question 4 (10 Marks)

Scenario 1:
Server registers with local Broker system

i. Broker is started in inialization phase of system. Broker enters event loop and waits for messages
ii. User, or some other entity, starts server application. Server executes initialization code. Server registers with broker
iii. Broker receives registration request. Extracts information from message and stores in repository. Acknoledgment is sent
iv. Server enters main loop waiting for client requests

Scenario 2:
Client sends synchronous request to local server.

i. Client app started. Client invokes remote server's method.
ii. Client-side proxy packs parameters and other information in message to forward to local broker
iii. Broker looks up location of server in repository. Server local, broker forwards message to server-side proxy.
iv. Server-side proxy unpacks parameters and other information. Server-side proxy invokes appropriate message on server
v. After completion, server returns results to server-side proxy which packages it to broker
vi. Broker forwards message to client-side proxy
vii. Client-side proxy unpacks result and returns to client

Continued...

a. Based on Scenario 1 and 2, draw the design pattern described. **(2 Marks)**

b. What are the strength of the design pattern shown in Figure 4.1? **(4 Marks)**

Continued...

c. How would yellow pages brokering be useful when a service is requested?

(2 Marks)

d. When is it particularly useful to use the Broker Handle pattern in place of the Broker Forwarding pattern?

(2 Marks)

Continued...

Question 5 (10 Marks)

Consider an application used to simulate and study robots interaction. The following classes have been written for this application.

- IBehaviour (Strategy) - an interface that defines the behavior of a robot
- Concrete Strategies: AggressiveBehaviour, DefensiveBehaviour, NormalBehaviour; each of them defines a specific behaviour. In order to decide the action this class needs information that is passed from robot sensors like position, close obstacles, etc.
- Robot - The robot is the context class. It keeps or gets context information such as position, close obstacles, etc, and passes necessary information to the Strategy class.
- In the main section of the application, several robots are created and several different behaviours are created. Each robot has a different behavior assigned: 'Big Robot' is an aggressive one and attacks any other robot found, 'John v.2.1' is really scared and run away in the opposite direction when it encounter another robot and 'D2' is pretty calm and ignore any other robot. At some point, the behaviours are changed for each robot.

a. What is the suitable design pattern to be used in this application? Draw a class diagram that can be used to implement this design pattern using the information provided in this scenario.

(6 marks)

Continued...

b. Explain the differences between the Factory Method and the Abstract Factory design patterns in terms of the *products* created AND *how* products are created. Draw the **class diagram** for a generic **Factory Method** and **Abstract Factory design patterns** as well.

(4 marks)

End of Paper